USER NEWS

CRAY BLITZ - world computer chess title contender

How does CRAY BLITZ do it? Does it have a positive mental attitude that just can't be beat? Or is it just nimbler and shrewder than most? Whatever it is, the prodigious program is making its mark. In CRAY CHANNELS Vol. 4 No. 3, we reported that CRAY BLITZ drew the final match against BELLE of Bell Laboratories in the 1982 world chess tournament sponsored by the Association for Computing Machinery. Bob Hyatt, who has been competing with CRAY BLITZ since 1980, and Harry Nelson at Lawrence Livermore Laboratory refined the program for 1982 and are fine-tuning it for the 1983 tournament. We thought we'd find out exactly what they've been up to with our favorite chess program, CRAY BLITZ.

Speed and smarts are key factors in the performance of a computer chess program. CRAY BLITZ has the speed, processing an average of 10,000 nodes/sec. (At last report, its closest competitor, CDC 4.9, was operating at 2,700 nodes/sec.) But Bob and Harry figured that if CRAY BLITZ could run even faster, the extra time could be used to analyze each move with even greater depth. That would give the program an additional edge against BELLE. While BELLE (a computer hard-wired exclusively for chess) can look at about 160,000 nodes/sec, it is not a particularly smart program. Enhancements to CRAY BLITZ's intelligence coupled with greater speed give it a good deal of clout against BELLE. Revectorizing and recoding parts of the program in CRAY Assembly Language (CAL) were important factors in its 1982 performance.

Bob Hyatt explained that Harry Nelson found certain "hot spots" where the program was spending too much time. Vectorization and CAL-coding decreased the amount of time spent in each of those routines by a factor of 10 to 20. Of CRAY BLITZ's 10,000 lines of FORTRAN coding, 2,000 were designated to be coded in assembly language. Those 2,000 FORTRAN lines turned into 8,000 lines of CAL. Care was taken to ensure that there were no surprises when it really counted. Before entering the tournament, the changes underwent rigorous testing for months, virtually 24 hours a day. Overall, the program is now running about five times faster than before.

Some areas where the program was taking a lot of time were in the Move Generator, Exchange Evaluator and Check Analyzer routines. In the Move Generator, for instance, the program was spending a tremendous amount of time generating potential moves via a tree search. In this operation, potential move sequences are identified that will avoid the loss of a player. A tentative move is selected. The Move Generator then switches sides and analyzes the opponent's potential response moves from the new position. This analysis continues until a fixed depth is reached.

The Exchange Evaluator subroutine was also revectorized and coded in CAL. This routine constantly checks to see if a move to a particular square is safe. The Exchange Evaluator had been absorbing 20-30% of the program's total operation time.

Bob went on to relate that the program is constantly worried about whether its king is in check. In normal play it would spend about 25% of its time per move determining the king's safety. And in tactical situations, up to 50% of the allotted time could be spent in the Check Analyzer. Vectorization has also significantly reduced the amount of time spent in this routine.

Intelligence enhancements that have recently been added will augment the program's capabilities in 1983 play. Bob Hyatt explained one of these: "Most computer programs don't recognize the difference between a "good" and "bad" bishop. In the game of chess, a "good" bishop is one whose movements are not impeded by its own pawns. A "bad" bishop is one that is restricted in the moves it can make because its own pawns are on the same color square it is, thus blocking it. If the program can recognize this situation, it can either avoid or correct it. Until now, CRAY BLITZ couldn't do that. By adding a simple check to determine the potential for this situation, problems can now be avoided." Bob went on to say, "To my knowledge, no other chess program has this particular checking capability. It's a nice feature to have, but not as necessary to the game as others. The routine takes up valuable time that most other computer chess programs simply cannot spare."

"CRAY BLITZ also plays an excellent end game," Bob said. "When there are only a few pieces left on the board — like a couple of pawns and a king, it plays a game that is far superior to other programs."

Hyatt is hopeful that CRAY BLITZ will be able to run on the CRAY X-MP in this year's tournament. He explained that the move evaluation time should be cut in half by using the parallel power of the X-MP. "Without doing anything to the program we should see time improvements of about 30% with the faster clock period. But what we are really waiting for is the CRAY-2."

28th Mersenne Prime found

The CRAY computer will keep its place in The Guiness Book of World Records as the discoverer of the world's largest prime number. One Saturday morning several months ago, the CRAY nonchalantly printed out the message: $2^{86243} - 1$ is a prime number (containing 25,962 digits). This event was no small cause for celebration, although the CRAY had no way of realizing it. Over 600 hours of CRAY time divided among several systems had been committed to the search for the 28th Mersenne Prime. This marks the second time that a CRAY system has discovered a largest known Mersenne Prime. The 27th was found in 1979.

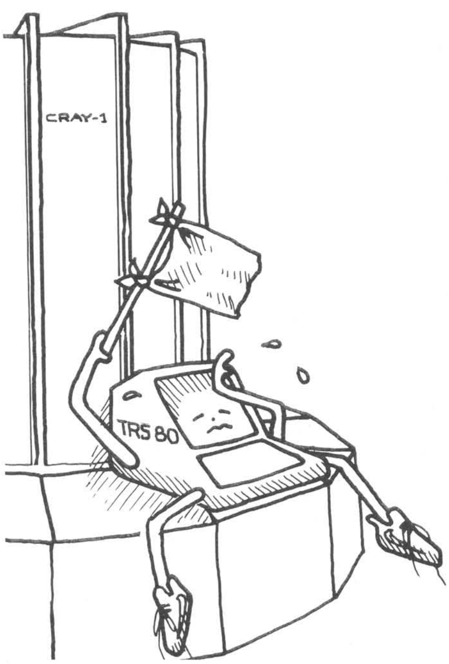
Many people have been involved in the efforts leading to the discovery of both numbers. Cray personnel, interested parties at several user sites and several CRAY systems have all taken part in the projects. Among them, Harry Nelson of Lawrence Livermore National Laboratory and David Slowinski, a Systems Analyst at Cray Research, have been most interested. Nelson and Slowinski collaborated on writing the program that found the 27th Mersenne Prime, and Slowinski was largely responsible for orchestrating the effort in finding the 28th.

The algorithm used to execute the Lucas-Lehmer test for primality was streamlined in the search for the 28th Mersenne Prime. Dave Slowinski explained, "About 90% of the time is spent squaring 25,000-digit integers. By optimizing squaring procedure in the program, I was able to improve performance by about 40% over the first program."

Slowinski had estimated that it would require 2,000 hours of computer time to hunt for the 28th Mersenne Prime, but with good luck and optimized code it was discovered in only 600 hours. To reduce the problem of finding the number to a workable scale, Slowinski divided the task among several CRAYs around the country, each one looking at some small part of the Mersenne sequence. The systems executed the program only when the computer was idle. Slowinski says that the program is also a useful confidence test for the hardware because it computes a residue check that catches many different kinds of failures.

For additional reading about David Slowinski's work with the 27th and 28th Mersenne Primes see CRAY CHANNELS, Vol. 4 No. 1, *"Searching, for the 27th Mersenne Prime"*, David Slowinski, pp. 15-17, and DISCOVER, Vol. 4 No. 2, February 1983, *"Biggest Prime, Longest Pi"*, Bruce Schechter, pp. 92-93.

A CRAY challenges a TRS-80



Why would anyone do it? Putting a lil' old TRS-80 just minding its own business up against a CRAY just isn't fair. And yet there are some people in the world who'll try just about anything. A Cray analyst who shall remain nameless is one of them. The gentleman benchmarked the performance of a TRS-80 Model 3 16K computer against the CRAY-1 S/2400, explaining that the unusual action was prompted by his son's school teacher who asked him to talk to the class about computers. He thought it would be fun comparing the performance of the school's microcomputer to the CRAY. He thought it would be something that the kids could relate to. Uh huh.

For his comparison, the analyst executed a 50 x 50 matrix multiply in both machines. Then he compared the time and cost of processing. The results are found below.

Machine	Execution Time (in seconds)	Per Unit Cost
TRS-80 Model 3	3180	$999
CRAY-1 S/2400	0.002	$7,620,000

Performance Ratio: 1/1,590,000
Cost Ratio: 1/7628

It was reported that screams of "Uncle, Uncle" could be heard from the school's computer room where the TRS-80 met its fate. The class was duly impressed with the speed of the CRAY. So was the TRS-80. The CRAY executed the problem approximately 1.6 million times faster than its distant cousin.

But you say, "Nobody expects the TRS-80 to be able to tackle a problem like that." We agree. However, there is a similarity between the systems in that they are both innovative wonders in their class. And the one basis for comparison may be the price/performance ratios of the systems. Based on the results, in order for the two to have equal price/performance ratios, the CRAY-1/S should cost 200 times more than it does, or the TRS-80 should cost 200 times less. While the classroom audience was not interested in that bit of trivia, we thought you might be.